

# Markov Chain Monte Carlo using the Hamiltonian method

*Kenneth M. Hanson*

Los Alamos National Laboratory

This presentation available under <http://www.lanl.gov/home/kmh/>

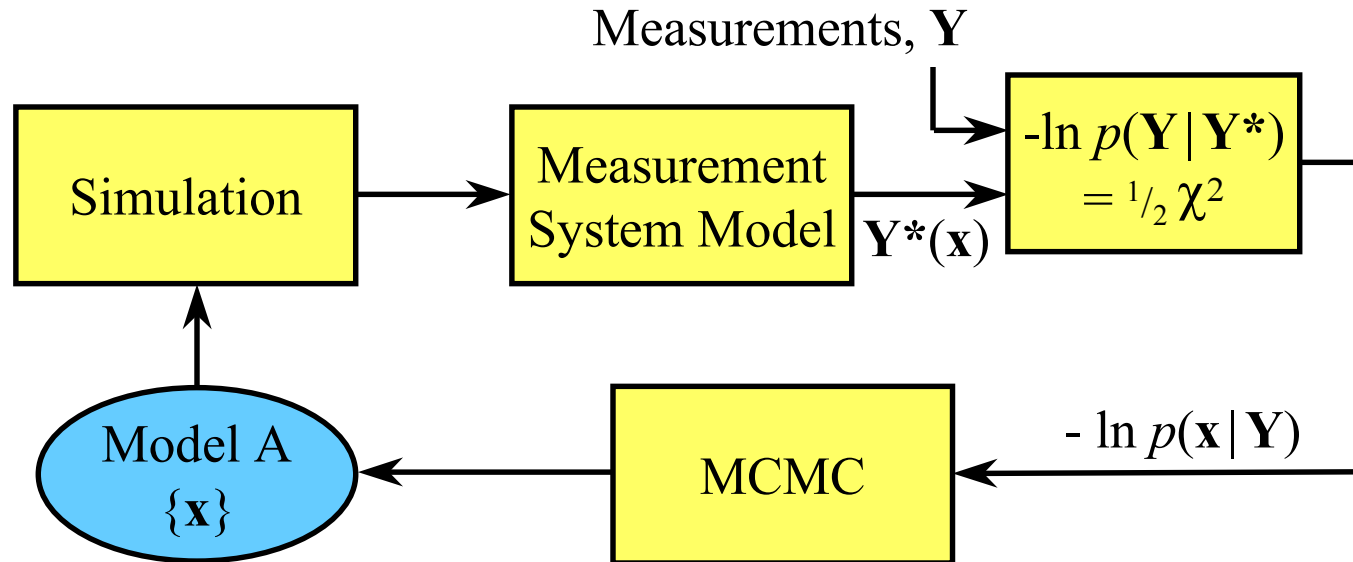
# Acknowledgements

---

- MCMC experts
  - Dave Higdon, Frank Alexander, Julian Besag, Jim Guberantus, John Skilling, Malvin Kalos
- General discussions
  - Greg Cunningham, Richard Silver

# MCMC in Bayesian data analysis

---



- - log(likelihood) distribution is result of calculation; function of model parameters  $\mathbf{x}$
- Markov Chain Monte Carlo (MCMC) algorithm draws random samples of  $\mathbf{x}$  from posterior probability  $p(\mathbf{x}|\mathbf{Y})$
- Produces plausible set of parameters  $\{\mathbf{x}\}$ ; therefore model realization

# MCMC - problem statement

---

- Parameter space of  $n$  dimensions represented by vector  $\mathbf{x}$
- Given an “arbitrary” **target** probability density function (pdf),  $q(\mathbf{x})$ , draw a set of samples  $\{\mathbf{x}_k\}$  from it
- Only requirement typically is that, given  $\mathbf{x}$ , one be able to evaluate  $Cq(\mathbf{x})$ , where  $C$  is an unknown constant, that is,  $q(\mathbf{x})$  need not be normalized
- Although focus here is on continuous variables, MCMC can be applied to discrete variables as well

# Uses of MCMC

---

- Permits evaluation of the expectation values of functions of  $\mathbf{x}$ , e.g.,

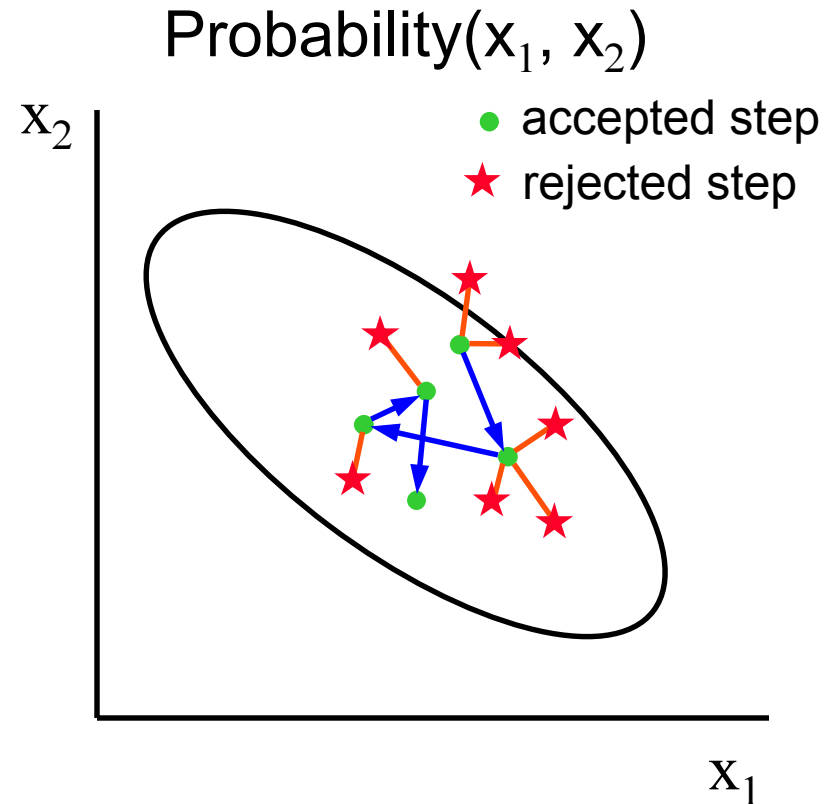
$$\langle f(\mathbf{x}) \rangle = \int f(\mathbf{x}) q(\mathbf{x}) d\mathbf{x} \cong (1/K) \sum_k f(\mathbf{x}_k)$$

- typical use is to calculate mean  $\langle \mathbf{x} \rangle$  and variance  $\langle (\mathbf{x} - \langle \mathbf{x} \rangle)^2 \rangle$
- Useful for evaluating integrals, such as the partition function for properly normalizing the pdf
- Dynamic display of sequences provides visualization of uncertainties in model and range of model variations
- Automatic marginalization; when considering any subset of parameters of an MCMC sequence, the remaining parameters are marginalized over (integrated out)

# Metropolis Markov Chain Monte Carlo

# Generates sequence of random samples from an arbitrary probability density function

- Metropolis algorithm:
  - draw trial step from symmetric pdf, i.e.,
$$t(\Delta \mathbf{x}) = t(-\Delta \mathbf{x})$$
  - accept or reject trial step
  - simple and generally applicable
  - relies only on calculation of target pdf for any  $\mathbf{x}$



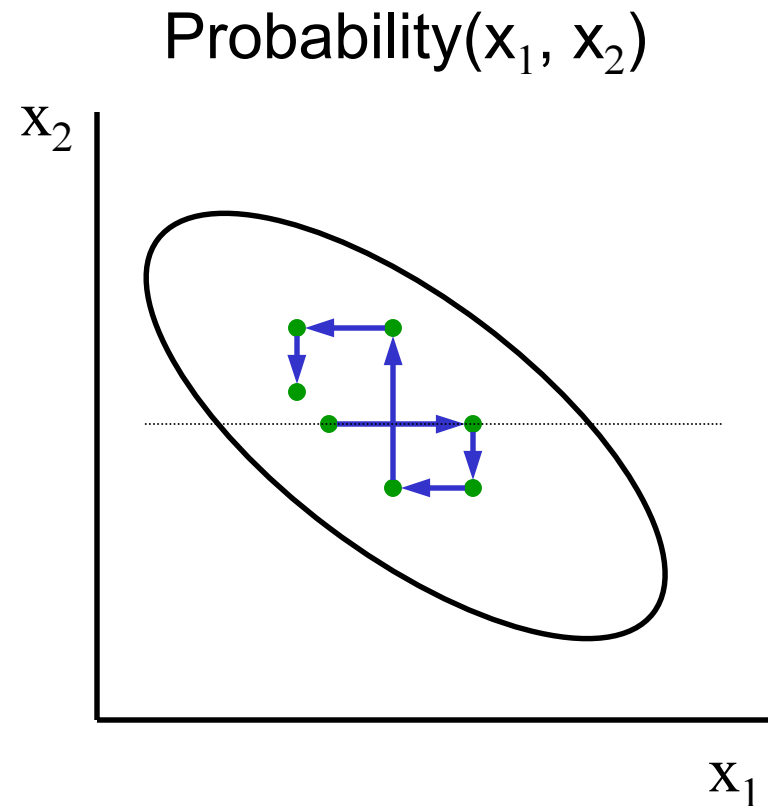
# Metropolis algorithm

---

- Select initial parameter vector  $\mathbf{x}_0$
- Iterate as follows: at iteration number  $k$ 
  - (1) create new trial position  $\mathbf{x}^* = \mathbf{x}_k + \Delta\mathbf{x}$  ,  
where  $\Delta\mathbf{x}$  is randomly chosen from  $t(\Delta\mathbf{x})$
  - (2) calculate ratio  $r = q(\mathbf{x}^*)/q(\mathbf{x}_k)$
  - (3) accept trial position, i.e. set  $\mathbf{x}_{k+1} = \mathbf{x}^*$   
if  $r \geq 1$  or with probability  $r$ , if  $r < 1$   
otherwise stay put,  $\mathbf{x}_{k+1} = \mathbf{x}_k$
- Requires only computation of  $q(\mathbf{x})$
- Creates Markov chain since  $\mathbf{x}_{k+1}$  depends only on  $\mathbf{x}_k$

# Gibbs algorithm

- Vary only one component of  $\mathbf{x}$  at a time
- Draw new value of  $x_j$  from conditional pdf
$$q(x_j | x_1 x_2 \dots x_{j-1} x_{j+1} \dots )$$
- Cycle through all components





# Hamiltonian method

---

- Often called **hybrid method** because it alternates Gibbs & Metropolis steps
- Associate with each parameter  $x_i$  a momentum  $p_i$
- Define a Hamiltonian (sum of potential and kinetic energy):

$$H = \phi(\mathbf{x}) + \sum p_i^2 / (2 m_i) \quad ,$$

where  $\phi = -\log (q(\mathbf{x}))$

- Objective is to draw samples from new pdf:

$$q'(\mathbf{x}, \mathbf{p}) \propto \exp(-H(\mathbf{x}, \mathbf{p})) = q(\mathbf{x}) \exp(-\sum p_i^2 / (2 m_i))$$

- Then set of samples  $\{\mathbf{x}_k\}$  represent draws from  $q(\mathbf{x})$ ;  
 $\mathbf{p}$  dependence marginalized out

# Hamiltonian algorithm

---

- Gibbs step: randomly sample momentum distribution
- Follow trajectory of constant  $H$  using leapfrog algorithm:

$$p_i(t + \frac{\tau}{2}) = p_i(t) - \frac{\tau}{2} \frac{\partial \varphi}{\partial x_i} \Big|_{\mathbf{x}(t)}$$

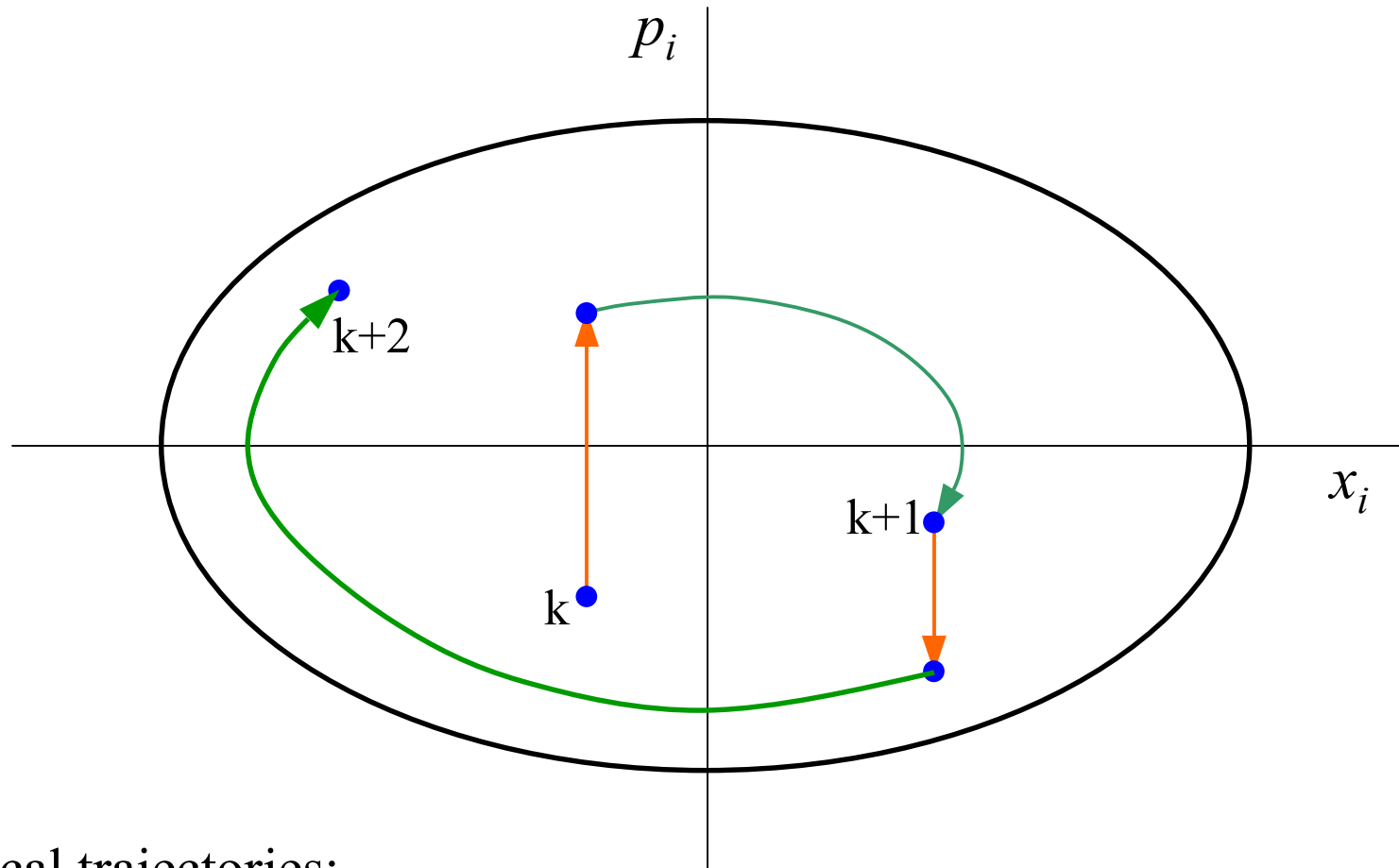
$$x_i(t + \tau) = x_i(t) + \frac{\tau}{m_i} p_i(t + \frac{\tau}{2})$$

$$p_i(t + \tau) = p_i(t + \frac{\tau}{2}) - \frac{\tau}{2} \frac{\partial \varphi}{\partial x_i} \Big|_{\mathbf{x}(t + \tau)}$$

where  $\tau$  is leapfrog time step

- Metropolis step: accept or reject on basis of  $H$  at beginning and end of H trajectory

# Hamiltonian hybrid algorithm



Typical trajectories:

red path - Gibbs sample from momentum distribution

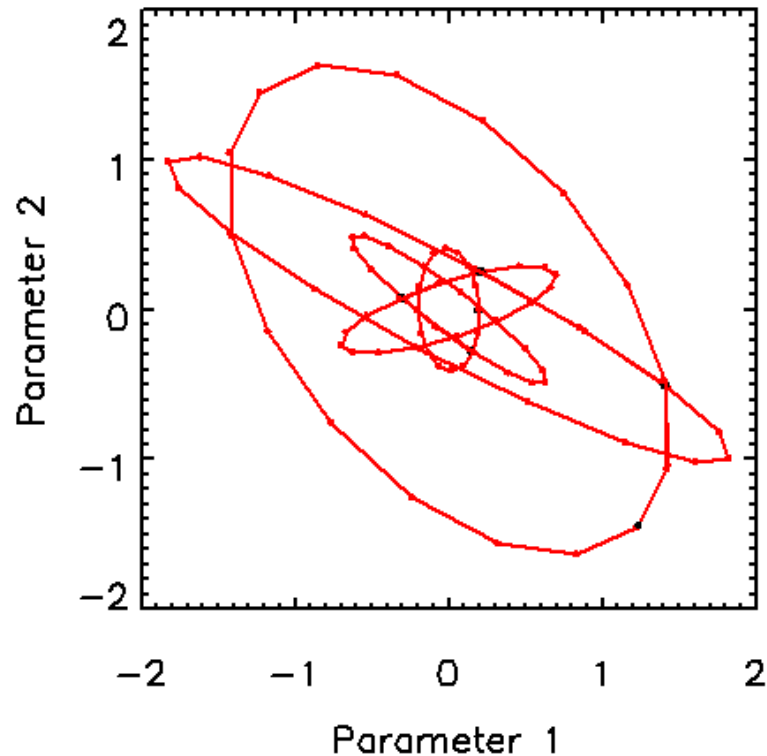
green path - trajectory with constant  $H$ , follow by Metropolis

# Hamiltonian algorithm

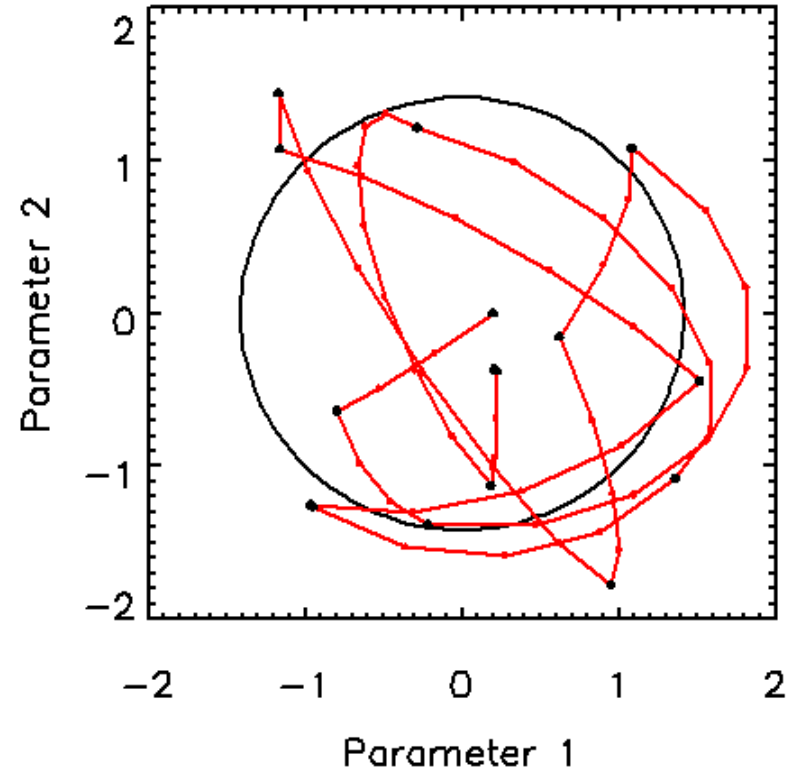
---

- Gibbs step - easy because draws are from uncorrelated Gaussian
- $H$  trajectories followed by several leapfrog steps permit long jumps in  $(\mathbf{x}, \mathbf{p})$  space, with little change in  $H$ 
  - specify total time =  $T$  ; number of leapfrog steps =  $T/\tau$
- Metropolis step - no rejections if  $H$  is unchanged
- Adjoint differentiation efficiently provides gradient

# 2D isotropic Gaussian distribution



Long H trajectories - shows ellipses  
when  $\sigma_1 = \sigma_2 = 1$ ,  $m_1 = m_2 = 1$



Randomize length of H trajectories  
to obtain good sampling of pdf

# MCMC Efficiency

---

- Estimate of a quantity from its samples from a pdf  $q(v)$

$$\tilde{v} = \frac{1}{N_k} \sum v_k$$

- For  $N$  independent samples drawn from a pdf, variance in estimate:

$$\text{var}(\tilde{v}) = \frac{\text{var}(v)}{N}$$

- For  $N$  samples from an MCMC sequence with target pdf  $q(v)$

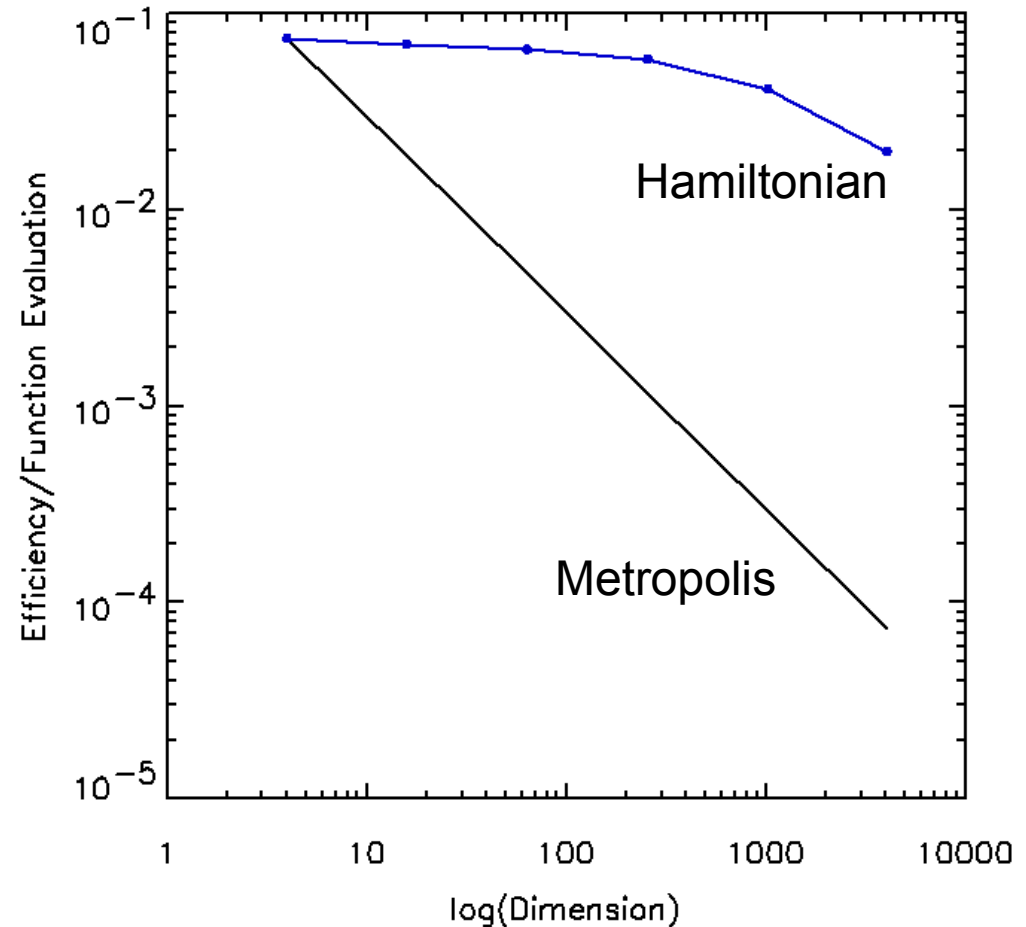
$$\text{var}(\tilde{v}) = \frac{\text{var}(v)}{\eta N}$$

where  $\eta$  is the sampling efficiency

- Thus,  $\eta^{-1}$  iterations needed for one statistically independent sample
- Let  $v = \text{variance}$  because aim is to estimate variance of target pdf

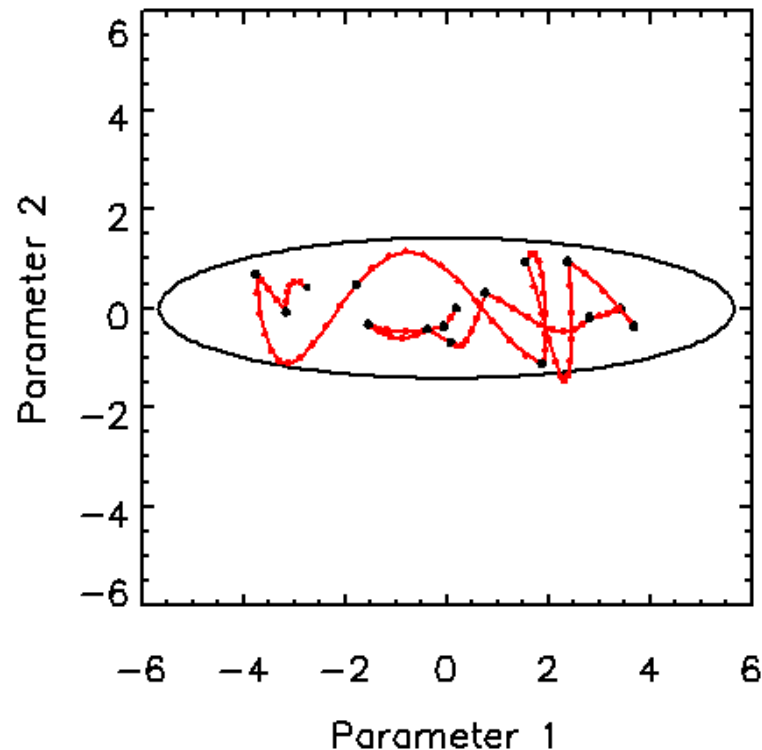
# n-D isotropic Gaussian distributions

- MCMC efficiency versus number dimensions
  - Hamiltonian method: drops little
  - Metropolis method: goes as  $0.3/n$
- Hamiltonian method much more efficient at high dimensions



# 2D nonisotropic Gaussian distribution

---



- Nonisotropic Gaussian target pdf:  $\sigma_1 = 4$ ,  $\sigma_2 = 1$ ,  $m_1 = m_2 = 1$
- Randomize length of H trajectories to get random sampling
- Convergence: determine whether sequence samples target pdf



# Convergence test statistic

---

- Variance integral

$$\begin{aligned}\text{var}(x_i) &= \int (x_i - \bar{x}_i)^2 p(\mathbf{x}) d\mathbf{x} \\ &= \int \frac{1}{3} (x_i - \bar{x}_i)^3 \nabla \varphi(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} + \frac{1}{3} (x_i - \bar{x}_i)^3 p(\mathbf{x}) \Big| \end{aligned}$$

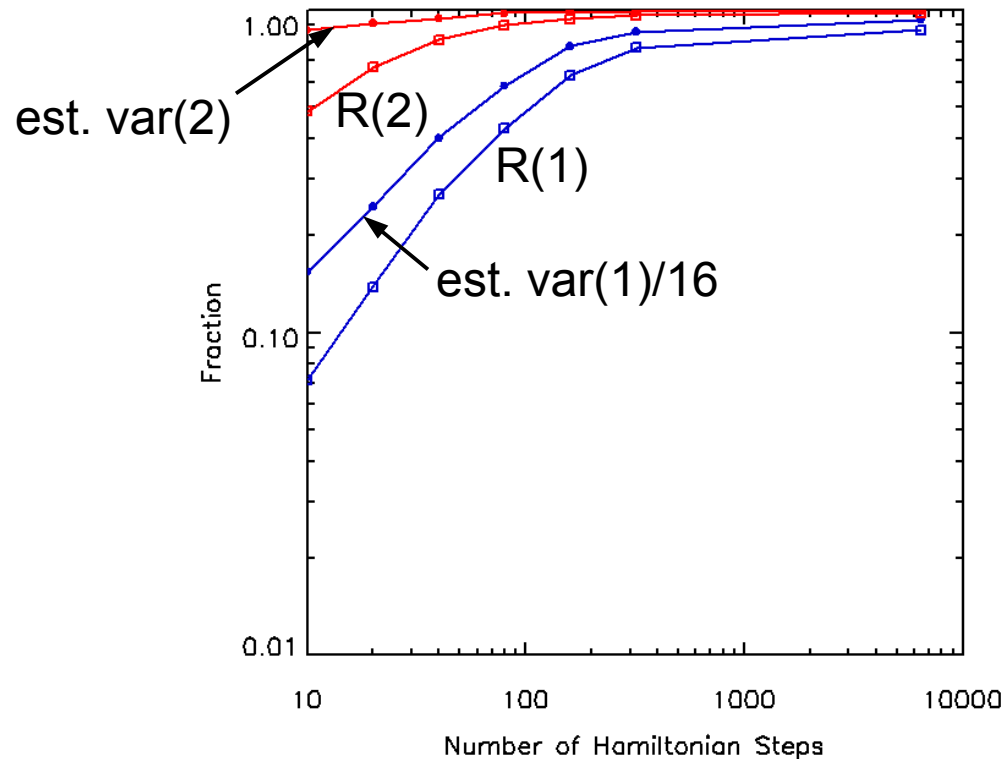
by integration by parts and  $\varphi(\mathbf{x}) = -\log(p(\mathbf{x}))$

- limits are typically  $\pm\infty$  and last term is usually 0
  - thus, integrals are equal
- Form ratio of integrals, computed from samples  $\mathbf{x}^k$  from  $p(\mathbf{x})$

$$R = \frac{\sum (x_i^k - \bar{x}_i^k)^3 \frac{\partial \varphi}{\partial x_i^k}}{3 \sum (x_i^k - \bar{x}_i^k)^2}$$

- $R$  tends to be less than 1 when  $p(\mathbf{x})$  not adequately sampled

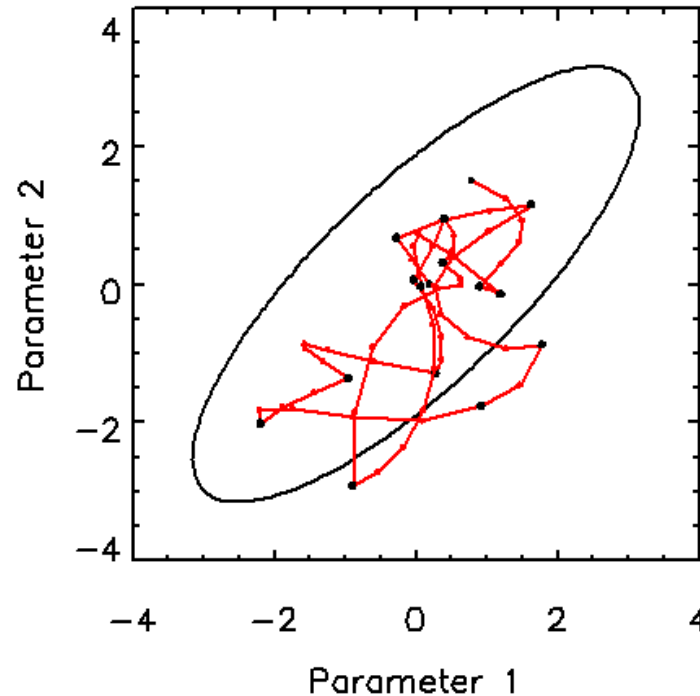
# Convergence - 2D nonisotropic Gaussians



- Nonisotropic Gaussian target pdf:  $\sigma_1 = 4$ ,  $\sigma_2 = 1$ ,  $m_1 = m_2 = 1$ 
  - control degree of pdf sampling by using short leapfrog steps ( $\tau = 0.2$ ) and  $T_{max} = 2$
- Test statistic  $R < 1$  when estimated variance is deficient

# 16D correlated Gaussian distribution

---



- 16D Gaussian pdf related to smoothness prior based on integral of L2 norm of second derivative
- Efficiency/(function evaluation) =  
2.2% (Hamiltonian algorithm)  
0.11% or 1.6% (Metropolis; w/o & with covar. adapt.)

# MCMC - Issues

---

- Identification of convergence to target pdf
  - is sequence in thermodynamic equilibrium with target pdf?
  - validity of estimated properties of parameters (covariance)
- Burn in
  - at beginning of sequence, may need to run MCMC for awhile to achieve convergence to target pdf
- Use of multiple sequences
  - different starting values can help confirm convergence
  - natural choice when using computers with multiple CPUs
- Accuracy of estimated properties of parameters
  - related to efficiency, described above

# Conclusions

---

- MCMC provides good tool for exploring the Bayesian posterior and hence for drawing inferences about models and parameters
- Hamiltonian method
  - based on Hamiltonian dynamics
  - efficiency for isotropic Gaussians is about 7% per function evaluation, independent of number of dimensions
  - much better efficiency than Metropolis for large dimensions
  - more robust to correlations among parameters than Metropolis
- Convergence test based on gradient of  $-\log(\text{probability})$

# Bibliography

---

- *Bayesian Learning for Neural Networks*, R. M. Neal, (Springer, 1996); Hamiltonian hybrid MCMC
- “Posterior sampling with improved efficiency,” K. M. Hanson and G. S. Cunningham, *Proc. SPIE* **3338**, 371-382 (1998); Metropolis examples; includes introduction to MCMC
- *Markov Chain Monte Carlo in Practice*, W. R. Gilks et al., (Chapman and Hall, 1996); excellent review; applications
- “Bayesian computation and stochastic systems,” J. Besag et al., *Stat. Sci.* **10**, 3-66 (1995); MCMC applied to image analysis
- “Inversion based on complex simulations,” K. M. Hanson, *Maximum Entropy and Bayesian Methods*, G. J. Erickson et al., eds., (Kluwer Academic, 1998); describes adjoint differentiation and its usefulness for solving inversion problems

More articles and slides under <http://www.lanl.gov/home/kmh/>